



1/22

FIG. 1  
PRIOR ART

1	c:\collections
2	notes.txt
3	myletter.doc
4	c-myhomepage
5	
6	s
7	homepage.html
8	myphoto.jpg

FIG. 2

1	c:\collections
2	notes.txt
3	myletter.doc

4	c-myhomepage
5	collspec
6	s
7	homepage.html
8	myphoto.jpg

100

FIG. 3

1	collection	c-myhomepage
2	coll-type	cf-web-page
3	coll-home	cf-colls:mysite.com:c-myhomepage
4	coll-desc	A sample homepage collection
5	end-collection	

102



2/22

FIG. 4

1 c:\collections  
2 programs  
3 helloworld  
4 c-hello-library  
5 c-hello  
  
6 c-myprogram  
  
7 parts  
8 c-include-files  
9 c-library-one  
10 c-library-two  
  
11 webstuff  
12 c-myhomepage

FIG. 5

1 # Pathnames showing filesystem location of collections  
2  
3 c:\collections\programs\helloworld\c-hello-library  
4 c:\collections\programs\helloworld\c-hello  
5 c:\collections\c-myprogram  
6 c:\collections\parts\c-include-files  
7 c:\collections\parts\c-library-one  
8 c:\collections\parts\c-library-two  
9 c:\collections\webstuff\c-myhomepage

FIG. 6  
PRIOR ART

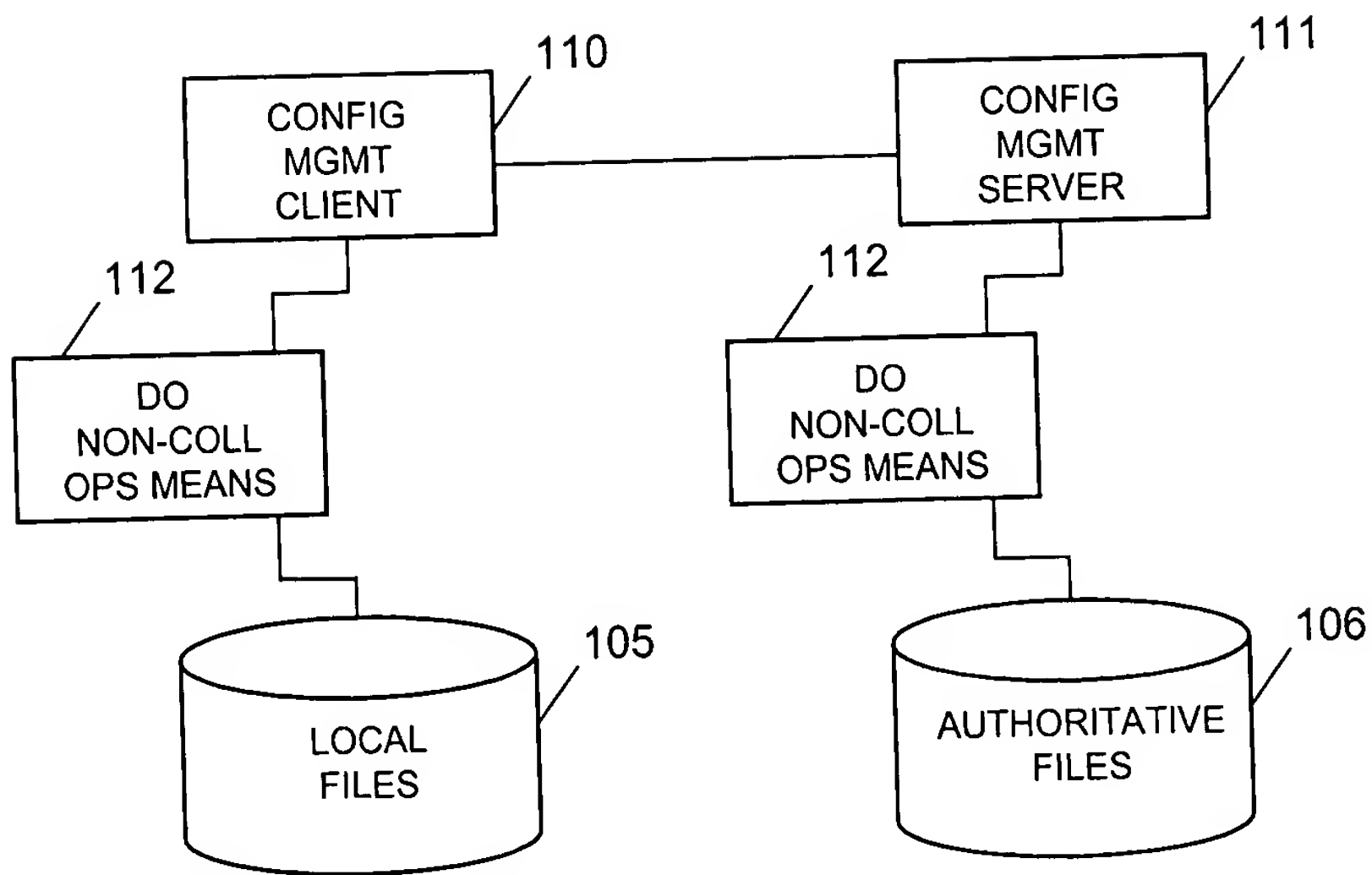
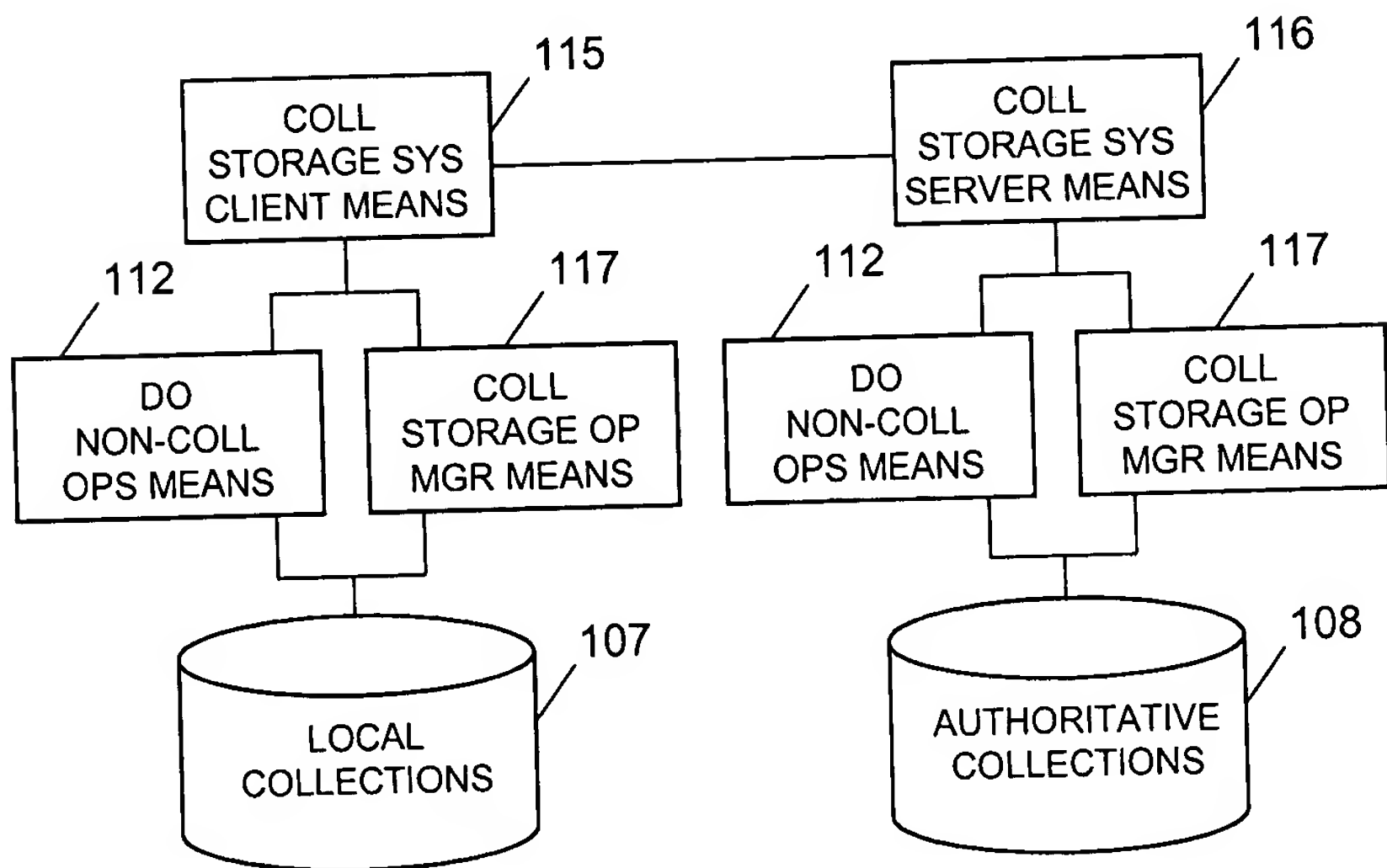


FIG. 7



+

4/22

## FIG. 8

```
1 # Five components of a full collection reference
2 #
3 # <coll-name><scope arguments><content selector names>
4 # where <coll-name> = category:authority:collection
5 #
6 category    the hierarchical category name of the coll
7              e.g. site/prod/release4
8 authority    the authority name responsible for managing the coll
9              e.g. colls.mysite.com
10 collection  the collection name
11              e.g. mycoll
12 scope-args  command line arguments that select scope within coll
13              e.g. -recursive -new -changed -locked
14 selectors   names of categories, directories, or files
15              e.g. mydir, mydir/myfile.c, collspec.txt
```

## FIG. 9

```
1 # Example collection reference for collection of FIG 2
2
3 # coll c-myhomepage in cat cf-colls at mysite.com
4 cf-colls:mysite.com:c-myhomepage
```

+



5/22

FIG. 10

1	# Shortcut collection references and their meanings	
2	#	
3	# Shortcut	Meaning
4		
5	cat:auth:coll	a full collection name reference
6	cat:auth:	all collections in category at authority
7	cat::coll	this coll in this cat at default authority
8	cat::	all colls in this cat at default authority
9	:auth:coll	this coll in all cats at this authority
10	:auth:	all colls in all cats at this authority
11		
12	::	current coll if inside a coll; invalid outside a coll
13	::.	current coll and current dir if inside; invalid outside
14	::mydir	current coll and mydir if inside; invalid outside





6/22

## FIG. 11

```
1  # Structure of a collection symbolic job request
2  #
3  # "do this (task) to that (collection reference)"
4  # <task-name> <collection-ref>
5  # where <collection-ref> = category:authority:collection
6  #
7  category    the hierarchical category name of the coll
8              e.g. site/prod/release4
9  authority    the authority name responsible for managing the coll
10             e.g. colls.mysite.com
11  collection  the collection name
12             e.g. mycoll
```

## FIG. 12

```
1  # Example collection symbolic job requests
2  # (using collections from FIGs 2-3)
3
4  # rebuild coll c-myhomepage in cat cf-colls at mysite.com
5  rebuild cf-colls:mysite.com:c-myhomepage
6
7  # rebuild all collections in category cf-colls at mysite.com
8  rebuild cf-colls:mysite.com:
```





7/22

FIG. 13

```
1 # Expanded collection list for a symbolic job request
2 #
3 # <task-name> <collection-ref>
4 # rebuild cf-colls:mysite.com:
5 #
6 cf-colls:mysite.com:c-hello-library
7 cf-colls:mysite.com:c-hello
8 cf-colls:mysite.com:c-myprogram
9 cf-colls:mysite.com:c-include-files
10 cf-colls:mysite.com:c-library-one
11 cf-colls:mysite.com:c-library-two
12 cf-colls:mysite.com:c-myhomepage
```

FIG. 14

```
1 # Expanded (sorted) visit order list for a symbolic job request
2 #
3 # <task-name> <collection-ref>
4 # rebuild cf-colls:mysite.com:
5 #
6 cf-colls:mysite.com:c-include-files      10
7 cf-colls:mysite.com:c-library-two       49
8 cf-colls:mysite.com:c-hello-library     50
9 cf-colls:mysite.com:c-library-one       50
10 cf-colls:mysite.com:c-hello             100
11 cf-colls:mysite.com:c-myhomepage        100
12 cf-colls:mysite.com:c-myprogram         100
```





8/22

FIG. 15

```
1 # Expanded platform list for cf-colls:mysite.com:c-myprogram
2 #
3 win2000
4 win98
5 win95
6 linux2
```

FIG. 16

```
1 # Expanded platform list for cf-colls:mysite.com:c-myhomepage
2 #
3 win2000
```

FIG. 17

```
1 # Expanded job triplet list for a single collection
2 #
3 cf-colls:mysite.com:c-myprogram      100  win2000
4 cf-colls:mysite.com:c-myprogram      100  win98
5 cf-colls:mysite.com:c-myprogram      100  win95
6 cf-colls:mysite.com:c-myprogram      100  linux2
```

FIG. 18

```
1 # Expanded job triplet list for a single collection
2 #
3 cf-colls:mysite.com:c-myhomepage 100  win2000
```





9/22

FIG. 19

```
1  # Expanded triplet list for a symbolic job request
2  #
3  # rebuild cf-colls:mysite.com:

4  cf-colls:mysite.com:c-include-files      10    win2000
5  cf-colls:mysite.com:c-include-files      10    win98
6  cf-colls:mysite.com:c-include-files      10    win95
7  cf-colls:mysite.com:c-include-files      10    linux2

8  cf-colls:mysite.com:c-library-two        49    win2000
9  cf-colls:mysite.com:c-library-two        49    win98
10 cf-colls:mysite.com:c-library-two        49    win95
11 cf-colls:mysite.com:c-library-two        49    linux2

12 cf-colls:mysite.com:c-hello-library      50    win2000
13 cf-colls:mysite.com:c-hello-library      50    win98
14 ...                                     ...    ...
15
16 cf-colls:mysite.com:c-library-one        50    win2000
17 cf-colls:mysite.com:c-library-one        50    win98
18 ...                                     ...    ...

19 cf-colls:mysite.com:c-hello             100    win2000
20 cf-colls:mysite.com:c-hello             100    win98
21 ...                                     ...    ...

22 cf-colls:mysite.com:c-myhomepage         100    win2000

23 cf-colls:mysite.com:c-myprogram          100    win2000
24 cf-colls:mysite.com:c-myprogram          100    win98
25 cf-colls:mysite.com:c-myprogram          100    win95
26 cf-colls:mysite.com:c-myprogram          100    linux2
```

10/22

FIG. 20

```
1  /* data structure for holding expanded job info */
2  csje-info {
3      + original symbolic task name
4      + original collection-reference
5
6      + expanded-coll-list
7
8      + coll-structure-1
9          + coll-name
10         + visit-order
11         + platform-list
12         + platform-1
13         + platform-2
14         + platform-...
15         + other-coll-info
16
17     + coll-structure-2
18         + coll-name
19         + visit-order
20         + platform-list ...
21         ...
22
23     + coll-structure-3
24         ...
25
26     + other expansion info
27 }
```

11/22

FIG. 21

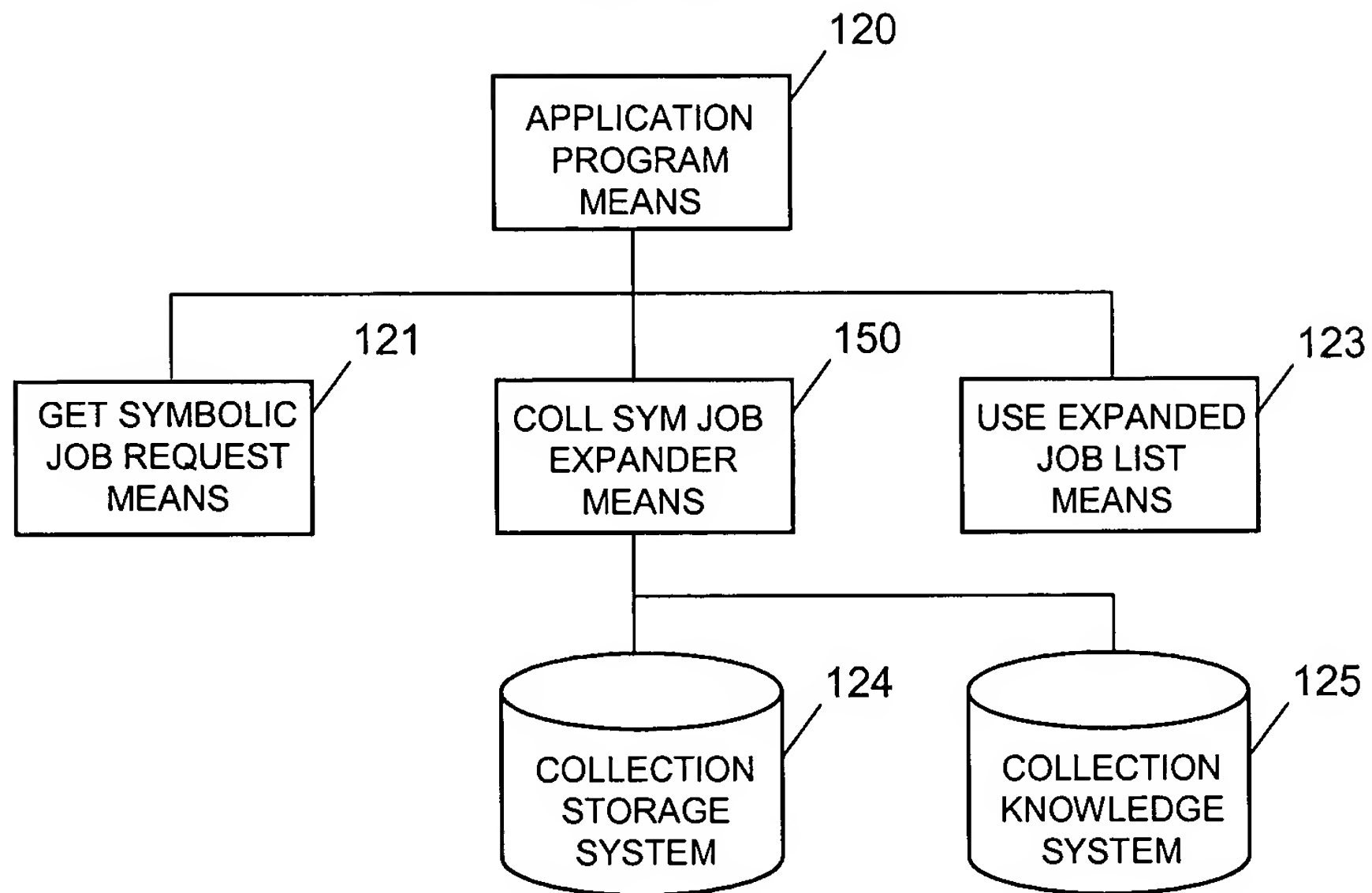


FIG. 22

- 1 # simplified algorithm for a program using a coll job expander
- 2 Call get symbolic job request means to get a symbolic job request
- 3 Call collection job expander means to expand symbolic request
- 4 Call use expanded job list means to use triplets on expanded job list

12/22

FIG. 23

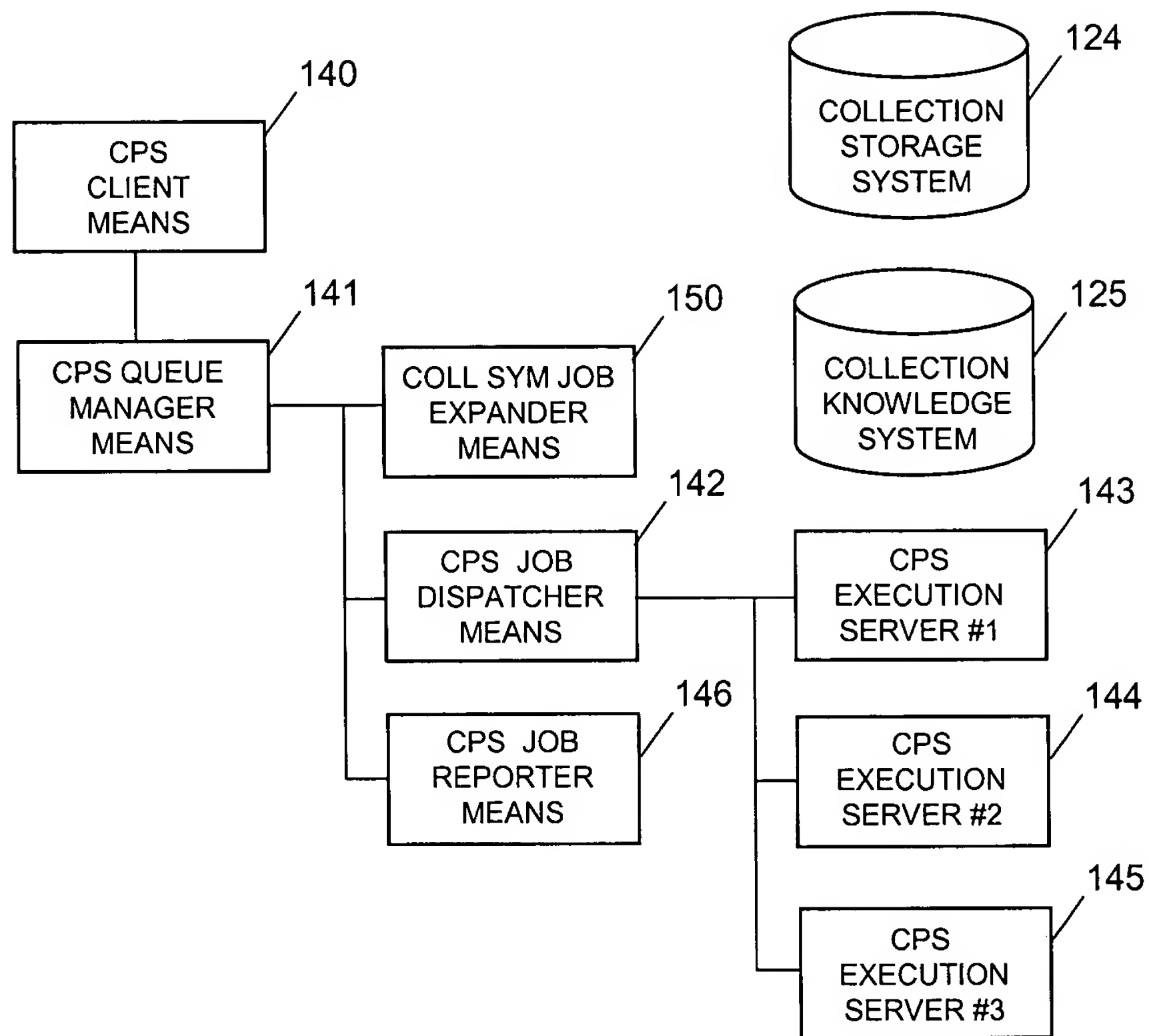


FIG. 24

- 1 # Simplified algorithm for a CPS queue manager means in a
- 2 # collection processing system
- 3 Receive collection symbolic job request from CPS client
- 4 Call collection job expander means to expand symbolic job request
- 5 Call job dispatcher means to execute individual expanded jobs
- 6 - job dispatcher distributes individual job triplets among N servers
- 7 Call collection job reporter means to aggregate and report job results

FIG. 25

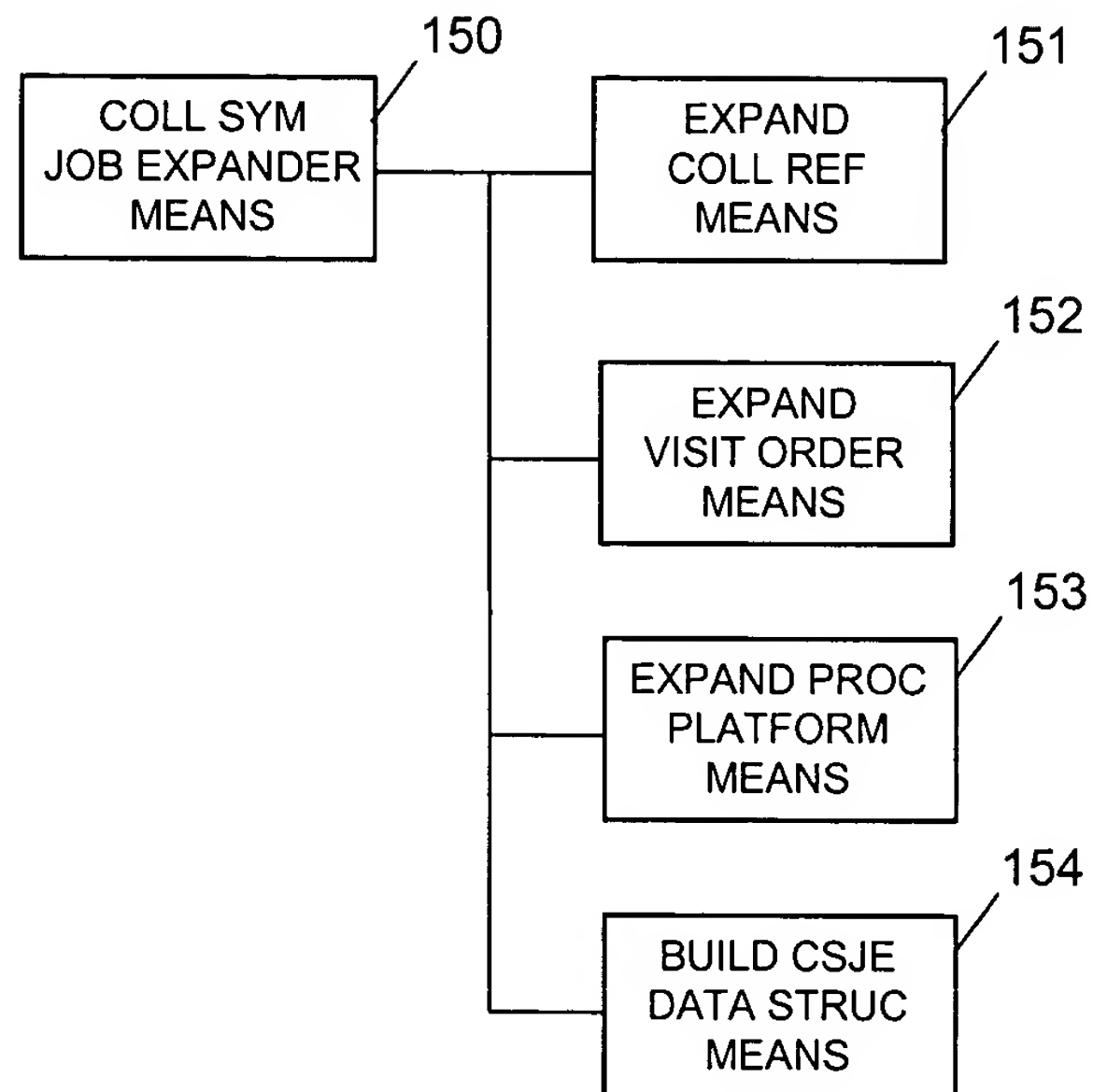


FIG. 26

- 1 # Simplified algorithm for a collection symbolic job expander means
- 2 #
- 3 Receive collection symbolic job request from caller
- 4 Call expand coll reference means to get list of individual collections
- 5 Call expand visit order means to get a visit order value for each coll
- 6 Call expand platform list means to get list of platforms for each coll
- 7 Build collection job expander info data structure to return to caller

14/22

FIG. 27

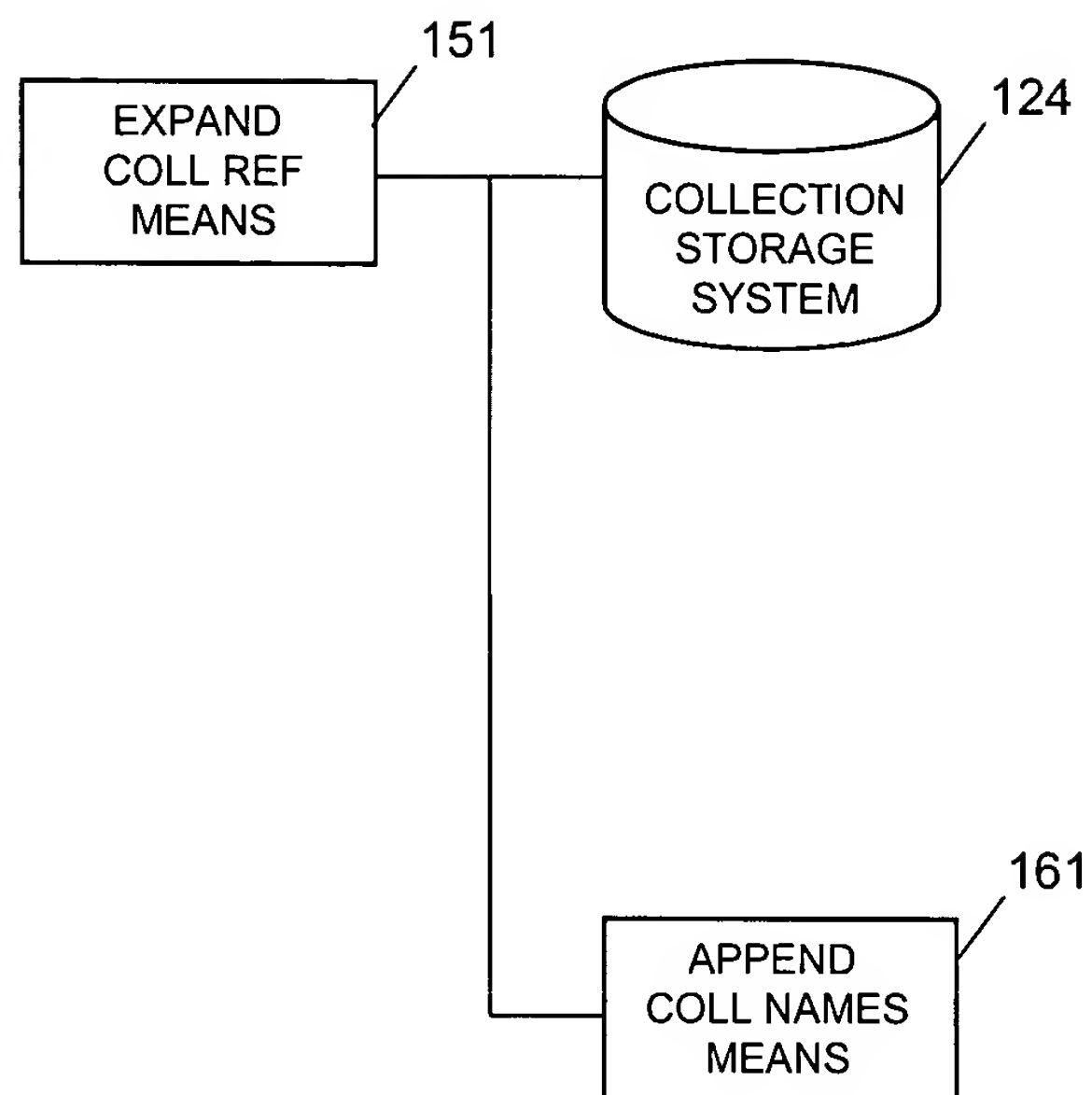


FIG. 28

- 1 # Simplified algorithm for an expand collection reference means
- 2 #
- 3 Receive unexpanded collection reference from caller
- 4 Call collection storage system to get list of individual collection names
- 5 Build csje-info data structure containing list of individual collection names
- 6 Return new data structure to caller

FIG. 29

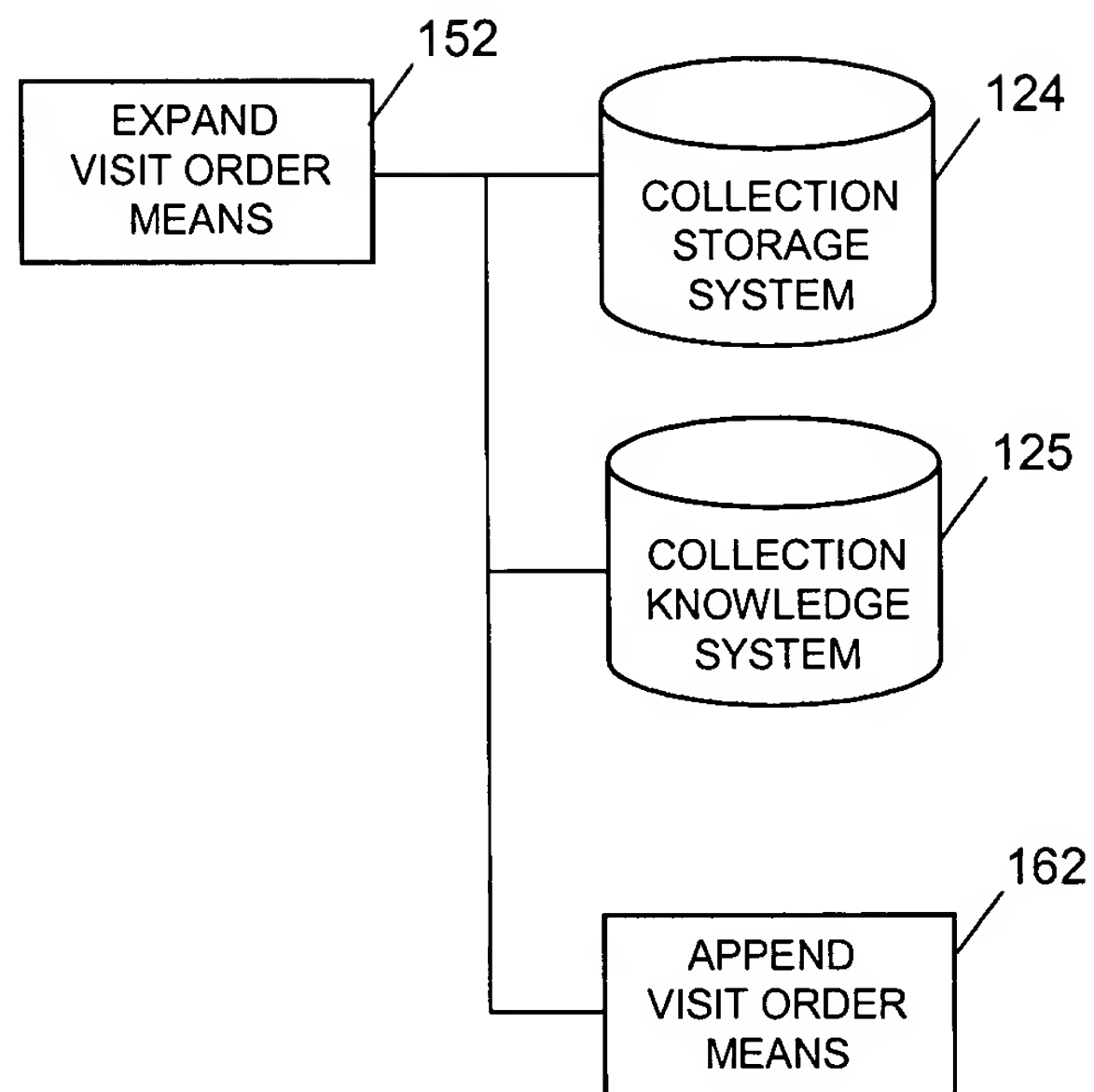


FIG. 30

- 1 # Simplified algorithm for an expand visit order means
- 2 #
- 3 Receive list of individual collection names from caller
- 4 Call collection storage system to get explicit visit orders for colls on list
- 5 Call collection storage system to get collection types for colls on list
- 6 Call collection knowledge system to get type definitions for colls on list
- 7 Use explicit visit order vales in preference to coll-type-def visit orders
- 8 Append coll types and proper visit orders to csje-info data structure
- 9 Return augmented data structure to caller



16/22

### FIG. 31

```
1 name-coll-type.tbl:
2 # a table of collection type names
3 # name          definition file
4 ct-program-c    ct-program-c.def
5 ct-library-c    ct-library-c.def
6 ct-doc-html     ct-doc-html.def
```

### FIG. 32

```
1 ct-program-c.def:
2 # a collection type definition file for a C program collection
3
4 # the processing platform type for all c-program collections
5 proc-platform-type    pp-program-c
6
7 # the default visit order type for c-program collections
8 visit-order           vo-program-c
```

### FIG. 33

```
1 ct-library-c.def:
2 # a collection type definition file for a C library collection
3
4 # the processing platform type for all c-library collections
5 proc-platform-type    pp-program-c
6
7 # the default visit order type for c-program collections
8 visit-order           vo-library-c
```







17/22

FIG. 34

```
1  name-visit-order-type.tbl:
2  # a table of visit order type names and values
3
4  # vo-type-name          vo-value
5  vo-initial              10
6  vo-include-files        20
7  vo-library-c            50
8  vo-program-c            100
9  vo-program-fortran      100
10 vo-doc-html             100
```

FIG. 35

```
1  collspec:
2  # a collection specifier with explicit numeric visit order value
3
4  collection
5  coll-home               cf-colls:mysite.com:c-library-two
6  coll-type               ct-library-c
7  ...
8  visit-order             49
9  end-collection
```

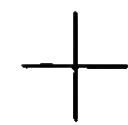




18/22

FIG. 36

1	c:\collections	
2	programs	
3	helloworld	
4	c-hello	ctype = ct-program-c default vo=100
5	c-hello-library	ctype = ct-library-c default vo=50
6	c-myprogram	ctype = ct-program default vo=100
7	parts	
8	c-include-files	ctype = ct-include-files default vo=10
9	c-library-one	ctype = ct-library-c default vo=10
10	c-library-two	ctype = ct-library-c <u>EXPLICIT vo=49</u>
11	webstuff	
12	c-myhomepage	ctype = ct-doc-html default vo=100





19/22

FIG. 37

1	# table of unsorted visit order values for list of colls	
2	# coll-name	visit-order-value
3	cf-colls:mysite.com:c-hello	100
4	cf-colls:mysite.com:c-hello-library	50
5	cf-colls:mysite.com:c-myprogram	100
6	cf-colls:mysite.com:c-library-one	50
7	cf-colls:mysite.com:c-library-two	49
8	cf-colls:mysite.com:c-include-files	10
9	cf-colls:mysite.com:c-myhomepage	100

FIG. 38

1	# table of sorted visit order values for list of colls	
2	# coll-name	visit-order-value
3	cf-colls:mysite.com:c-include-files	10
4	cf-colls:mysite.com:c-library-two	49
5	cf-colls:mysite.com:c-hello-library	50
6	cf-colls:mysite.com:c-library-one	50
7	cf-colls:mysite.com:c-hello	100
8	cf-colls:mysite.com:c-myprogram	100
9	cf-colls:mysite.com:c-myhomepage	100

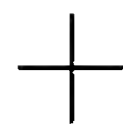


FIG. 39

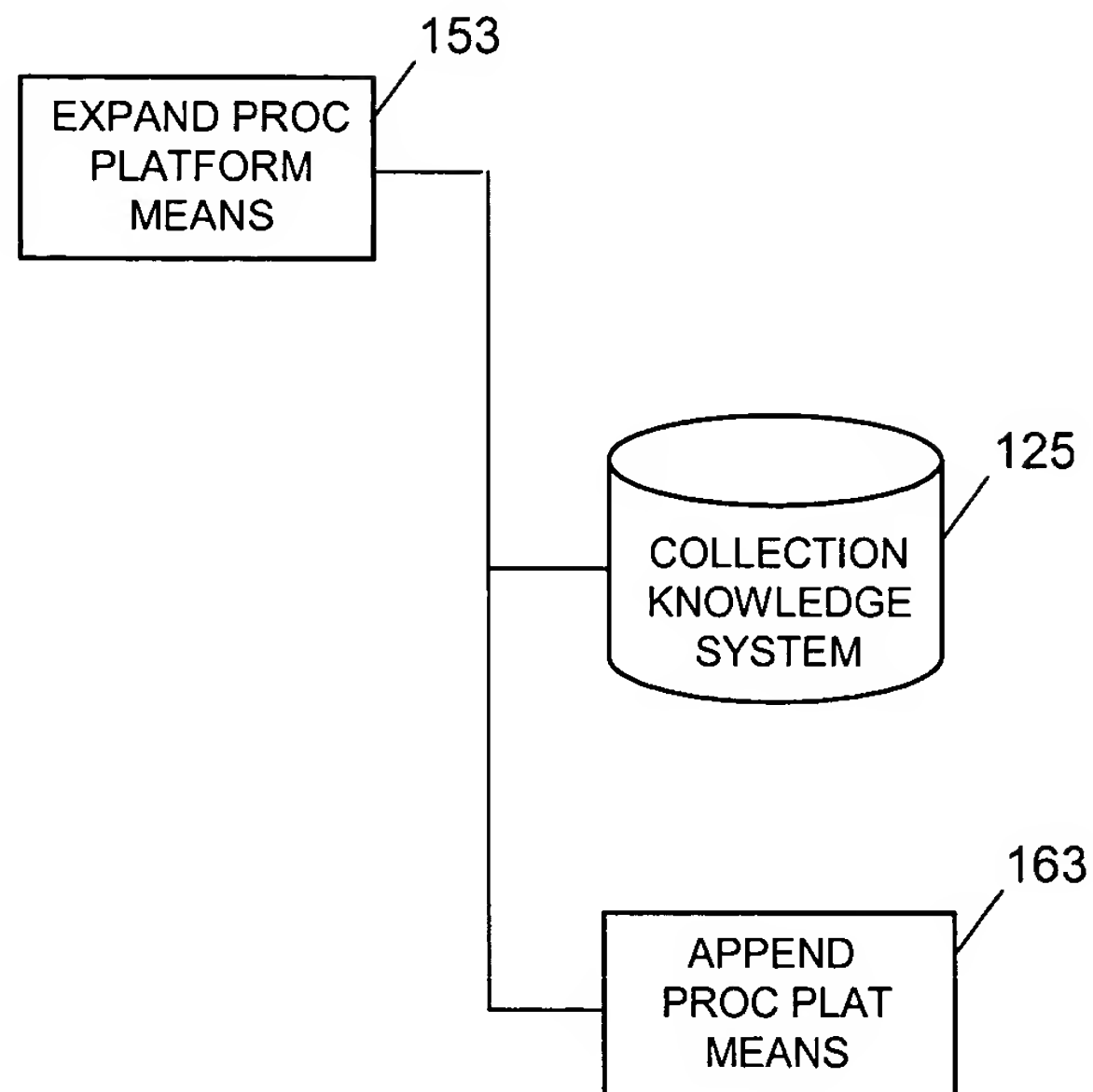
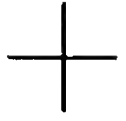


FIG. 40

- 1 # Simplified algorithm for an expand processing platform means
- 2 #
- 3 Receive list of individual collection names and coll types from caller
- 4 Call collection knowledge system to get exe platform types for coll types
- 5 Call collection knowledge system to get exe platform type definitions
- 6 Append list of exe platforms to each coll in csje-info data structure
- 7 Return augmented data structure to caller



21/22

FIG. 41

```
1  name-proc-plt.tbl:
2  # a table of collection processing platform types
3
4  # name                definition file
5  pp-program-c          pp-program-c.def
6  pp-program-fortran    pp-program-fortran.def
7  pp-web-page           pp-web-page.def
```

FIG. 42

```
1  pp-program-c.def:
2  # a processing platform type definition file for a C program
3
4  # process c program collections on these platforms
5  win2000
6  win98
7  win95
8  linux2
```

FIG. 43

```
1  pp-web-page.def:
2  # a processing platform type definition file for a web page
3
4  # process web page collections on these platforms
5  win2000
```

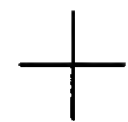


FIG. 44

